Table indexes

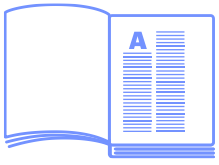
Primary, Foreign, and User Indexes

GeneXus™

Indexes

Database index

Book index



The concept of a "database index" is similar to the concept of a "book index" which allows you to access specific content without having to go through all the pages of the book.

In a database, then, an index is a structure that provides quick access to the records of a table, increasing the speed of operations and allowing you to search data without having to run through the entire table sequentially.

For example:

CUSTOMER

CustomerId	CustomerName
1	Jonathan
2	Alexander
3	Peter
4	Susan
5	Christopher
6	Ann

CUSTOMER
Index by CustomerName

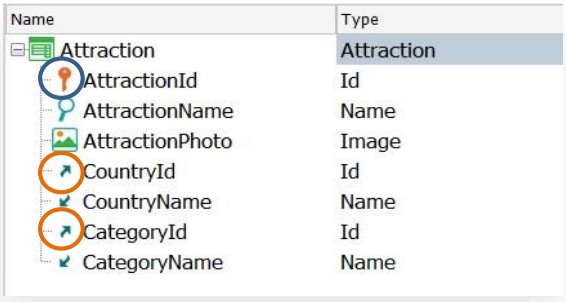
CustomerId	CustomerName
2	Alexander
6	Ann
5	Christopher
1	Jonathan
3	Peter
4	Susan

Consider, for example, the Customer table. If you want to search for the record corresponding to the name Christopher, you must run through each row until you find it. On the other hand, if you have an index defined by CustomerName, you can access the record directly without having to run through the whole table sequentially.

Now that you've learned the concept of index, note what indexes are automatically created by GeneXus and how developers can create their own indexes as needed.

Indexes automatically created by GeneXus.

Attraction Transaction



ATTRACTION Table

GeneXus automatically creates:

- Primary index
- Foreign index

GeneXus automatically creates indexes that not only allow efficient access to the tables but also perform efficient referential integrity checks.

Consider the structure of the Attraction transaction.
•Note that AttractionId is the primary key.
•CountryId and CategoryId are foreign keys.

In the KB Explorer window, below the transaction itself you can see the associated table. When opening it, you can also see the Indexes tab that shows the indexes automatically created by GeneXus. All these indexes have a name, the attribute by which that index is defined, and their order. While it can be ascending or descending, it is ascending by default.

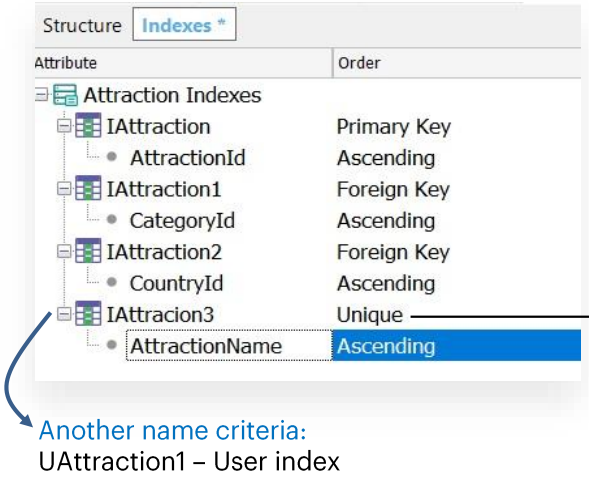
Primary index: This index called IAttraction over the primary key AttractionId is used to control the uniqueness of the record; that is, that there aren't two tourist attractions with the same identifier. But it also controls, for example, that at the moment of creating a tour to that attraction, the AttractionId value indicated in the tour previously exists as the primary key in the ATTRACTION table.

Primary index is used to:
- Efficiently perform reference integrity checks.
- Make referential integrity checks when inserting or modifying a foreign key value.

GeneXus automatically defines all primary indexes for the transactions' identifying attributes.

Next, look at the IAttraction1 and IAttraction2 foreign indexes over the foreign keys CategoryId and CountryId, respectively. These foreign indexes are also automatically created by GeneXus, and are used to efficiently perform reference integrity checks.
For example, if you want to delete the category with identifier 2, this index checks that there are no related records in Attraction with the value CategoryId = 2.

User indexes: Duplicate / Unique



What if you need to make a list of tourist attractions in alphabetical order? It would be very convenient to have an index over the name of the attractions.

So, define a user index by the AttractionName attribute.

User indexes are, therefore, indexes defined by the developer to perform different searches efficiently.

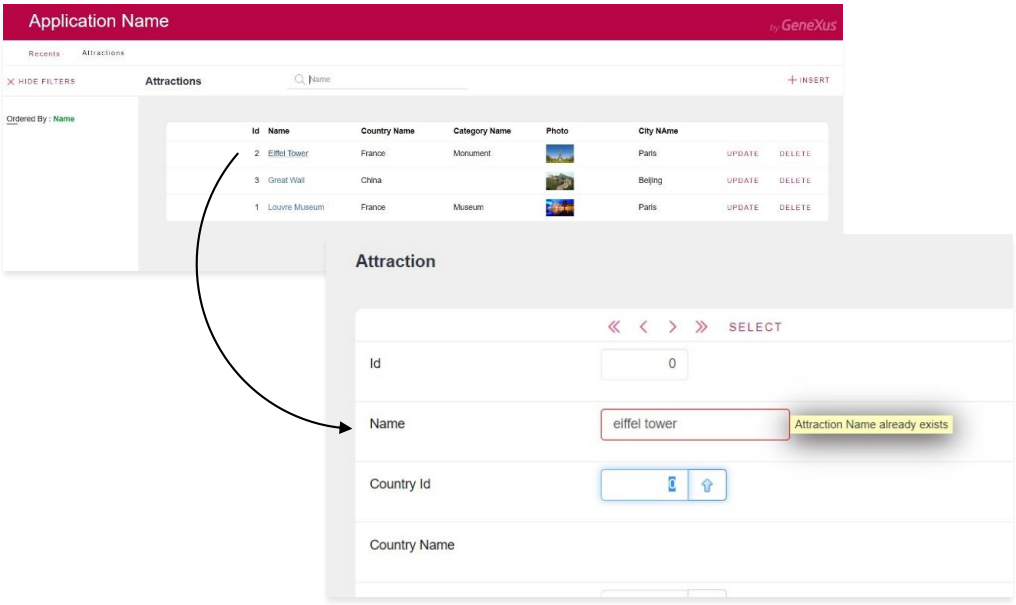
You may wonder: Why aren't these indexes also automatically created by GeneXus? Simply because they are not necessary for making referential integrity checks. Remember that it is possible to sort a table by any attribute, or set of attributes, when necessary.

An index will accept duplicates when its values can be repeated. Think, for example, of the names of people that can be repeated. If you need an index to define an alphabetical order by the names of the people, this index must accept duplicate values; people's names can be repeated.

If, on the other hand, you indicate that the index is Unique, you will be defining that the value of that attribute, or set of attributes by which the index is being defined, cannot have a repeated value. GeneXus will automatically control the uniqueness of its value.

In this case, it will be said that the attribute, or the set of attributes by which the index is being defined, is a **candidate key**, since GeneXus will control its uniqueness as it does with the primary key, although it is not the primary key of the transaction.

At runtime...



Note, for example, that the Eiffel Tower has been entered.

When trying to enter a new tourist attraction called Eiffel Tower,
regardless if you type it in lowercase... GeneXus automatically controls the uniqueness of the attraction name, and displays a message indicating that the name already exists.

Likewise, by defining the corresponding unique indexes, we can control that no categories or countries with the same name are entered.

1-1 relationship

TripId	TripDate	SupplierId
1	10/10/2020	2
2	10/11/2020	1
3	12/12/2020	4

SupplierId	SupplierName
1	First Supplier
2	Second Supplier
3	Third Supplier
4	Fourth Supplier

Unique index by SupplierId

Candidate key

Suppose that a tour is run by only one provider, and one provider runs a single tour. So, there is a 1-1 relationship between both entities.

How can this be represented in GeneXus?
We can think of a 1-1 relationship as a particular case of a 1-N relationship, right? Because N represents “many” and, in particular, “many” includes 1.
So, the Provider's identification attribute is added as a foreign key in the Tour, although it could be done the other way around as it is a 1-1 relationship.

It must be checked that the value of SupplierId is not repeated in Trip, and thus make sure that a tour or trip has a single provider, and that one provider is in charge of only one tour. How can it be done?
By defining a unique index over the foreign key SupplierId in TRIP.

In this way, the SupplierId attribute will be a **candidate key in Trip**, and GeneXus will automatically control its uniqueness.

We will return to the 1-1 relationship later.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications